Supplementary File for the paper: Theorising the take-up of ICT: Can Valsiner's three zones framework make a contribution?

Michael Hammond and Bader Alotaibi

Overview

This support file is divided into three sections:

Summary report of the findings from the survey and interviews

In the account that follows key 'facts' of the data are bracketed, e.g. [1], [2], [3] and these match the codes used in the paper itself.

What is the context in which lecturers use ICT in the teaching of university-level mathematics?

With regard to access to ICT, lecture halls were all equipped with blackboards (or whiteboards), overhead multimedia projectors and e-podiums (lecterns). An Internet connection was available for all staff and students. Learning management systems were set up. Lecturers had their own office space and most had portable technology [1]. The majority of survey respondents felt that it was not difficult for them to schedule a class in a computer lab [1], though a sizeable minority disagreed [2]. Most felt they had adequate technical support, but around a quarter disagreed [2]. In addition to mathematical and statistical software packages general programming languages (e.g. Fortran, C, C++ and Visual Basic) and Microsoft Excel were available. The majority of the respondents reported that they had good access to the software packages they needed for teaching [1]. Further, most respondents reported participating in training workshops on the use of technology in teaching. These workshops were held in mathematics departments, in the IT services departments or at outside venues. However, the workshops were mainly focused on introducing 'generic' e-learning, including the use of VLEs, and not specifically tailored to the use of mathematical software. [3]

In talking about teaching, interviewees explained that they felt they had considerable autonomy over teaching styles and little in the way of direction on how to teach [4]. However they taught to a predetermined syllabus and while they could input into the process, what was taught and how it was assessed could only be amended with the approval of the relevant committee [4]. In practice most classes followed a traditional lecture format with in some courses further practice of techniques in smaller workshop sessions.

How, and to what extent, do lecturers use ICT in the teaching of university-level mathematics courses?

The findings serve to reinforce a general observation that the take-up of ICT is under-developed for teaching and learning. Lecturers used software mainly in computational mathematics and statistical courses, while in others, and particularly in calculus courses, software was used only occasionally. Mathematica, Matlab and Maple were the three most frequently used items of software in more general mathematics courses while Minitab, Statistica and SAS were the three most commonly used items of statistical software. Around half of respondents reported using the presentation software PowerPoint when presenting their lectures and a slightly lower number reported using the smart board when teaching. [5]

The use of the VLE was quite high, with only around a quarter of lecturers reporting not having used it. The most frequent use of the VLE was to publish lecture notes and post support material such as past exam papers. Lecturers primarily used VLEs to support teacher led communication, whereas activities that encouraged collaboration or reflection, such as journals and wikis, were rarely promoted [5].

There was diversity in terms of take-up: users were divided between low users, mid users and high users. Low users used ICT in less than quarter of their lectures and offered students little out of classroom encouragement to use ICT, some did not use ICT at all for teaching and learning purposes. At the other end of the scale around a fifth were high users of ICT [6]. Amongst these were lecturers who had created archives of web resources for students, started up blogs and in some cases led workshops showing the use of mathematical software to colleagues. In many cases these lecturers were proactive, for example searching out for themselves freeware blogs to use with students and putting in time to support student activity [6].

Do particular groups of mathematics lecturers use ICT more than others?

The key finding was that lecturers who taught statistical and computational mathematics courses used software more, and were more likely to assign homework that required the use of software, than lecturers who taught theory courses [7]. This was rather an unexpected finding and one that has not been well reported in the literature. However the association was clear. Interviews showed that statisticians and computational mathematicians were teaching courses that required the use of software and were scheduled at times to teach in laboratories [7]. In contrast, pure mathematicians were less likely to be users of software as they focused on abstract and mostly theoretical concepts. Software use was not ground into the very notion of 'doing' pure mathematics in the same way as, say, SPSS is into 'doing' statistics and the assessment of data handling [7]. Descriptive statistics, reinforced that subject identification was a more important 'factor' in the take-up of ICT than other variable, though of course many factors were inter linked. For example, those identified clearly in the survey as holding constructivist views about teaching were more likely to focus their teaching on problem-solving activities and were less likely to be users of software. In contrast those identified as holding instructionist views mostly taught non-computational specialists and were more likely not to use ICT. The point here is that influence of pedagogic beliefs were strongly mediated by subject specialism and context.

What encourages/motivates lecturers to use ICT?

The major encouraging factors were both internal to the individual lecturer and external or environmental. First there was a widely held belief that the use of software had epistemic value - it was good for 'doing' mathematics. Interviewees gave a wide range of examples in which software could enable learners to carry out calculations very quickly or produce graphical displays that would be impossible or very difficult to do otherwise [8]. For example using software learners could visualize geometric shapes, especially in 3D spaces, or could better focus on understanding concepts rather than on procedural calculation. The use of software enabled mathematical concepts to be represented in different ways.

Several lecturers went on to speak about the affective value of ICT and most felt that students appeared more engaged and interested when using ICT, while about two thirds of the respondents to the questionnaire thought that using mathematical software in teaching increased 'interactivity' [8]. Contemporary software was also valued for being particularly easy to use especially when compared to more 'traditional' programming languages such as Fortran or C++ [9].

Lecturers thought that heads of schools and university managers were 'pushing' the use of e-learning. In this sense they felt they had been encouraged to use ICT [10]; however, direction and, in some ways, support was limited. A tradition of academic freedom gave lecturers the choice to teach either with or without ICT [4] and implementation was largely seen as laissez-faire [10].

What discourages/constrains lecturers from using ICT?

While there was a great deal of support for the proposition that mathematical software had a positive impact on students' learning there was a sizable minority who were more sceptical or at times simply disagreed [11]. For example, just under a third of respondents felt the use of software in the teaching of undergraduate mathematics did more harm than good. As an example some lecturers explained that it was more natural to use 'chalk and talk' for this created an appropriate pace in teaching - students needed time to absorb what the problems were about. It was also felt that students could become over reliant on the software and that software may be used as merely 'pressing buttons' to provide answers. Some felt, particularly at the undergraduate level, the use of software might be counterproductive. Students needed to make the effort and spend more time learning mathematics from first principles before resorting to ready-made functions. Challenge was important and productive [11].

In terms of the immediate environment lecturers were aware of an encouragement to use technology but there were mixed messages [12]. For example the use of software was encouraged but in many cases such use was not assessed and in examinations students were only allowed to use basic calculators (i.e. hand-held calculators without programming capabilities) [12]. Many felt that using ICT was infeasible when there was little time to deviate from a content heavy and inflexible curriculum [13]. While there was generally agreed to be good access to technology a constraint lay in access to adequate technical support and appropriate training on how to use mathematical software [14]. Furthermore, other than the computer for overhead projection, there were no computers in the teaching rooms, no internet connection and students did not tend to bring their own machines to class [15]. It was of course possible to use a computer laboratory but this required special effort unless timetabled, as was the case for, say, teaching statistics [7]. Some were able to put their reluctance to make an extra effort into using technology into a wider context. They reported that in their departments the focus was more on research output rather than teaching [7] and hence only a few of their colleagues were enthusiastic about using ICT. Furthermore some saw students as a conservative influence on their teaching which dampened their desire to innovate [16]

The interviews

During university visits interviews were conducted with nine mathematics and statistics lecturers. They were chosen as they represented all of the main research interests in mathematics. Table 1 shows the distribution of interviewees at each university according to their specialisms within mathematics.

Table 1: The interviewees and their specialisms

Specialism	University (A)	University (B)
Pure Mathematics	Two lecturers (A1, A2)	Two lecturers (B1, B2)
Applied Mathematics	One lecturer (A3)	Two lecturers (B3, B4)
Computational Mathematics	Two lecturers (A4, A5)	Two lecturers (B5, B6)
Statistics	Four lecturers (A6, A7, A8, A9)	Three lecturers (B7, B8, B9)

The findings are organized around three main categories: use of software, rationale for using software and rationale for not using software.

Use of software

At university (A), despite the availability of ICT facilities including mathematical software, mathematics lecturers used software only in courses that required its use, of which there were only two courses: numerical analysis and computational mathematics. The only exception was one faculty member, who used e-learning, including the use of mathematical software, in all of his courses. His specialization was computational mathematics. He explained that every week the students had a forum; self-assessment; links to the topic of the week and to various questions; and a journal, where each student could write whatever they wanted to write. He said that his goal was for students to have fun learning. He acknowledged that the students still preferred the traditional way of teaching, and he realized that change towards learning that depends on ICT may need more time. He asserted that students should be allowed to learn mathematics within a 'smart' learning environment, and he stressed that the use of software would be a key factor within that environment to allow students' minds to open up and see the beauty of mathematics.

Unlike the situation at university (A), mathematical software packages (e.g. Mathematica, Maple, Minitab, MATLAB, Excel), and programming languages (e.g. Fortran, C++) were used more frequently in teaching undergraduate mathematics courses at university (B). Software packages were used as a mathematical aid in some of the lower-level courses such as calculus and algebra, but were not an essential part of these courses. Some lecturers reported using software during their presentations to illustrate some of the topics. For example, B6 emphasized that he used software only as a supplementary part and as a teaching tool. He added that the use of software was not specified in the course syllabus, with the exception of courses that required it, such as statistics and numerical analysis courses. He mentioned that taking a programming language course (mostly C++ or Fortran) was a core requirement for all students at the university. B1 stated that he used the package Mathcad in the teaching of a pre-calculus course. Although more lecturers may have used software in teaching in this department compared to the former department, it must be stressed that the use of chalk and board was still the dominant practice even in this department. As B6 asserted, one should use software as an aid, but not as a substitute to the traditional methods of teaching, particularly in a subject like mathematics.

There were degrees of flexibility and autonomy when it came to using software packages in the teaching of mathematics at both universities. Although a course description would specify all the topics and textbooks in each particular course, it was actually left to the lecturer to choose what an appropriate teaching tool, including software packages was. This was true even when teaching courses that required the use of software packages. For example, B5, who was a lecture of numerical analysis, used Excel, and not MATLAB, C + +, Fortran, or any other software that is commonly used in such a course. When asked about the reason for choosing Excel and not MATLAB or other software, he said that the reason was that by using Excel the user would be able to have complete control over the entire process from A to Z. He stressed that the aim of using software should not be a process of 'pressing buttons' without knowing what really happened inside the device, which makes many mathematicians, in his view, reluctant to use the software. He added that one should be careful when it comes to the use of software, especially at the stage of university mathematics, which should be the stage of building students’ mathematical backgrounds. In his opinion, the initial stages of undergraduate study are the construction phase of the students' backgrounds, and should not be impeded by the use of software packages.

The use of software is essential in statistical courses. All of the statisticians in the sample from university (A) emphasized that they used statistical packages in all statistical courses they were teaching. They stressed that in statistics, after collecting numerical data, the objective is to analyse the data using software, then one can make decisions based on that analysis. A6 stressed that because they were dealing with data they were probably the biggest users of software. He added that in mathematics, one could teach the majority of courses without the use of software, whereas most statistical courses are heavily dependent on data analysis, in which the use of software is essential.

Because of the large number of students at university (A), it was difficult to accommodate them all with what was available in the laboratories. The statisticians at this university stated that students from outside the Department of Statistics were not offered statistical laboratories. In other words, statistical lecturers at university (A) used statistical packages during their presentations. However, students were not able to use statistical packages because the university did not provide them with laboratory times in such courses. The only exception was those students who were majoring in statistics.

As was the case at university (A), statistical lecturers at university (B) used statistical software packages (e.g. STATISTICA, Minitab and SAS) heavily during their teaching. The only obvious difference between the two departments was that all students who enrolled in statistical courses at university (B) were offered statistical laboratory sessions, whether these courses were offered to statistics students or to non-statistics students. This was perhaps because the number of students at this university was much smaller than at university (A). B8 stressed that statistical analysis is an integral part of coursework in statistical courses; therefore, statistical courses were taught mostly in computer labs, where every student would be able to do their coursework and assignments using the software.

Types of software used

In term of the software packages used when teaching the undergraduate mathematics courses, Mathematica was the most widely used mathematical package with 25 references from 12 participants (out of 18) mentioning that they were using it in their lectures. MATLAB followed closely with 20 references from 11 different participants. The third-most frequently used mathematical software was Maple, which was mentioned in 13 different references by 10 participants in the sample. These three were, by far, the most frequently utilized mathematical software. Less-common mathematical software included Scientific WorkPlace, Excel, GeoGebra and Mathcad. More general software such as C, C++, Fortran and Visual Basic were also mentioned, but less frequently. Wolfram Alpha, an on-line environment that uses Mathematica, was mentioned by one participant who used e-learning in all of his classes. He created ‘cyber classes’ or webpages with links to on-line resources for use beyond the lesson: ‘… in the front page of the cyber class there is a link to Wolfram Alpha… it's beautiful for them to check anything they want. For example in integration they go to Wolfram Alpha if they want to see some questions they cannot find the answer for or whatever. They just write integral or 'INT' and they see immediately the answer with graphical solution and all sorts of things. So these kinds of software are beautiful for helping our students…so what I always hope is to encourage my students and for that I use all the facilities available. For example, Excel is a lovely environment and it is available for all students. MATLAB of course is used heavily…I prefer Maple sometimes over Mathematica' (A4).

All of the statisticians in the sample (seven lecturers) used software. This is not surprising as statistics depends heavily on the analysis of large data sets. All the seven statisticians always used statistical packages in teaching. Minitab was the most commonly used software, used by all of the statisticians (plus one mathematician) in the sample, with 13 references. SPSS (10 references), STATISTICA (7 references) and SAS (5 references) were also commonly used.

The courses in which software was used

Regarding the courses in which software was used, as shown in Table 2, computational mathematics courses, which include numerical analysis, were the most frequent mathematical courses in which software packages were used with 19 references from 8 participants. Calculus (either 2D or 3D) followed that closely with 15 references from 9 sources. These two mathematical courses were by far the most reported courses in which software was used. This is not surprising as computational mathematics by nature depends entirely on computation and programming. Calculus courses are probably one of the most offered university courses that are compulsory and a prerequisite for many courses at most universities.

Perhaps one of the most predictable results in this study was that statistical courses were the top of all of the courses in which software was used. There were 14 references in which 8 different sources mentioned that they taught statistical courses that involved the use of software. Examples of such courses included: statistics for engineering (3 references), statistics for management students (3 references), non-parametric statistics (2 references) and data and analysis of experiments (1 reference). In addition, software was used in a variety of other mathematical courses, such as: differential equations (5 references), linear algebra (2 references), abstract algebra (2 references) and applied mathematics (3 references).

Table 2: Example of courses in which software was used

Course	Number of references	Number of sources
Calculus	15	9
Pre-calculus	1	1
Calculus in 3D	2	2
Computational mathematics	19	8
Differential equations	5	4
Linear algebra	2	2
Abstract algebra	2	2
Applied mathematics	3	2
Statistics	14	8
Number theory	1	1
Analysis	1	1
Linear programing	1	1

Examples of taught topics in which software was used

A wide range of examples of statistical and mathematical topics were mentioned by the participants. For instance, in calculus courses, some interviewees reported that they used Mathematica or Maple to integrate, differentiate, find the limit of or graph a particular function. In the more advanced 3D calculus, some of the participants stated that they used software to help students visualize service convolution and to find the volume of a function. In numerical analysis, some participants said that they used MATLAB and Excel in topics such as matrices computations, to produce iterations and to write codes when applying different methods to find the roots of equations. In differential equations (e.g. ODE and PDE), some of the interviewed lecturers reported using software packages to solve differential equations, and once the solutions were found, to plot the resulting functions. In other mathematical courses, further examples mentioned by the participants included:

- To expand a function in a Fourier series in a Fourier analysis course
- To plot a vector field in and to animate a tangent vector in PDE
- To implement the Gauss-Jordan method to solve a system of linear equations in linear algebra

- To calculate the centre, centralizer, normalizer and normal subgroup in abstract algebra
- To visualize and rotate cross sections of cylinders to get an estimate of the volume in geometry
- To demonstrate the areas of a triple integral in 3D calculus
- To find the mean, median, standard deviation and variance of particular data
- To implement regressions of some observations in statistics.

Some participants mentioned situations that software could not handle well. For example, B6 pointed out that when handling topics such as singularities or 'when things get closer to zero', many of these software packages could not handle such problems correctly. Therefore, students were warned not to rely totally on the technology to come up with solutions in such cases.

Where software was used

There were considerable differences between mathematicians and statisticians. Computer laboratories were used mostly by statistics lecturers, and not by mathematics lecturers. However, statisticians used statistical packages in their lectures using the multimedia projectors as well as in the computer labs. In addition, students of statistics were required to do their coursework using software in the laboratory. In mathematical courses in which the use of software is required (i.e. in numerical analysis and computation mathematics), the participants stated that usually one or two lectures were allocated to allow the lecturers to explain to the students the basic commands of the chosen package, and then students were advised to use online help to get acquainted with such software. In such courses, the lecturer introduced the software in the context of teaching (usually using presentation software) then students were asked to do homework assignments that involved the use of software. In terms of assessment, statistics students were assessed in the laboratory because some of the questions required using statistical packages, while mathematics students were assessed through written examinations with no use of software. In computational mathematics courses though, there were homework assignments that required the use of software for which about 10% of the total marks were allocated (e.g. A5, B6).

Learning management systems

Using LMS were voluntary and mainly used by lecturers to make general announcements, to connect with students via e-mail or to post course-related materials. B4 stated that he used Bb in three ways: making general announcements to students, sending e-mails to groups or individual students and posting lecture notes, assignments, exams, quizzes, etc. 'Most of our faculty members have a website and they post various materials for their courses, for example, exams, lecture notes, syllabus for each of their courses' (A1). There were no online courses offered through the LMS. As A1 pointed out, it was not an easy task to build a course online because it required team work or at least someone who was a mathematician and an ICT expert at the same time. A4 indicated that there was a process of change. For example, a new committee had been set up to design e-courses.

A4 mentioned that he also worked with the National Centre of E-Learning and Distance Learning to develop e-courses such as calculus and numerical analysis to be accessed by lecturers and students in all universities in Saudi Arabia. A4, who was a self-motivated lecturer, indicated that he had his own online courses in the LMS Moodle before his university subscribed to Bb. The main objective of using LMS according to A4 was 'just to be connected with your students and to make a community of learning'.

At university (B), an online homework system for calculus had been developed. B3 stressed that he had found many benefits in using the e-assessment system. However, one drawback he had witnessed was that lecturers did not see the details of students' working. B3 added he usually asked his students to attach PDF files of their work in order to see how they had arrived at their answers. B3 claimed the online homework system was generally preferred over traditional paper-based examinations. Colleagues viewed it as an effective and efficient way to support learning. B3 stressed that the use of e-assessment reduced the chances of cheating and provided the students with opportunities to learn from their mistakes.

Now that we have seen how software packages were used for teaching by lecturers from different branches in the mathematics and statistics departments at the two universities, we turn to explore the motivating factors, and also the impeding factors that hindered the use of software.

Rationale for the use of software

The motivating factors will be divided into two parts: factors related to beliefs about the value of software for learning, teaching and motivation and contextual factors. First we will discuss factors related to beliefs about the value of software for learning, teaching and motivation.

Speed and automatic calculation

Interviewees gave a wide range of examples that suggested that software could enable lecturers and students to calculate things very quickly, or to produce displays that would be impossible or very difficult to do otherwise. This saved them a lot of time, and re-focused students' attention away from complicated calculations that would otherwise take most of the time for the lecture; this was especially important when performing such calculations was not the direct objective of the lecture. B3 pointed out that it was a waste of time every time a lecturer was faced with a function that needed to be expanded in a Fourier series, for example, to do that calculation manually. In such a situation, it may be helpful to use the software. B3 asserted that he only used software for routine problems or calculations that would take a long time, or when such calculations were impossible to do by hand. He gave an example of trying to plot a complicated graph in three dimensional spaces and how it was very difficult to draw such a graph properly without spending most of the lecture time trying to draw a decent graph. B2 questioned the value of spending a long time in solving every single problem in, say, integration or differentiation manually if the students had understood the concepts involved. He felt that software should be used to obtain 'quick answers' and that these answers can be used to discuss underlying conceptual issues.

Software assists visualization

Visualization was the second main purpose of the use software packages in the teaching of mathematics. For respondents, visualization meant that the images and geometric shapes that are produced by the software can help students to see (or visualize) geometric shapes, especially when things get complicated in three dimensions spaces. Visualizing mathematical objects can be static or animated. For example, as B6 put it: 'visualization of the functions, especially when you get to the topics of solids and convolutions, it is much easier with Mathematica to get students visualizes the whole solid and how to find the volume'. A2 explained that with software, lecturers can actually display the 3D shape from various angles and rotate it, turn it upside down and even select parts of it and zoom in to highlight that part, and so on. Without using software as a tool to assist visualization, A2 stated that many students would not have such opportunities to visualize 3D mathematical objects. He stressed that using software to assist visualization was a great blessing to the teaching community that helps to transfer knowledge more effectively. B2 did not hide his criticism of those lecturers who were totally opposed to the use of the software; he stated: 'instead of just doing the calculations on the board or graphing them or doing the calculation without really seeing the object we are calculating, it doesn't make any sense to me'. A4 stated that he used GeoGebra specifically because it helped the students to visualize geometric objects. Then he criticized mathematicians who focus more on an algebraic approach and neglect the visual side of mathematics. He said he was 'amazed' to see in a course like calculus, which deals with 'functions', students could not visualize even a 'constant function'. He did not blame this only on the higher education curriculum but he felt this went back to teaching in school. He mentioned the movement known as 'calculus reform' which has emphasized the use of the senses to 'visualize' mathematical functions and let students 'see' the functions not just represent them algebraically. He stressed that students should be given a learning environment which afforded visualizing mathematical functions. In such an environment, software should be an easily accessible object.

Speed automation and assisting visualization can both help to acquire the third objective, which was to make things clearer to students so they can understand and absorb things better, as we shall see below.

Software makes things clearer

As was indicated previously, mathematics involves two types of skills: procedural skills and intellectual skills. Clarifying things for students is a main concern and a desired goal for most teachers in mathematics courses. This goal can be achieved with the use software packages in several respects.

Some of the participants made it clear that the goal of making things clearer could be achieved by focusing more on understanding concepts rather than focusing too much on procedural calculation, which itself was not the direct goal of the lecture. This in turn would also save energy for students, allowing them to focus on more important things that were the immediate objectives of the lecture. A2 stated: 'why to consume students' energy if I am doing a work for an hour and that work can be done, say, within one minute. So why do I keep working for another 59 minutes? So it saved time and left him time to go into higher thinking activities'. In the same context, A4 mentioned that there was no need to waste precious time in the lecture performing boring procedural calculations, especially if they were not the immediate objectives of the lecture. Then he gave the following analogy of who would do that: 'instead of just spending the time in what was known in England as 'donkey work'… we have to move them away from that'.

Software could also aid the lecturer in representing mathematical ideas in multiple ways within the very limited time of the lecture (i.e. multiple representations), which could make things clearer for different types of learners or as A2 pointed out: 'one can use the software to demonstrate the same object in different ways'.

Some participants emphasized that the use of software would likely lead to understanding not only the use of software itself but also understanding the theoretical basis behind such use. Consequently, it could lead to understanding the underlying mathematical topics that involved such usage. In this regard, A2 stressed that if students used software, it would not only speed up their computations, but they would also develop understanding of the theory behind such use. Making things clearer was a desirable goal, even in a subject like statistics where the use of software is necessary, or as A8 stressed, in statistics, students' ideas are strengthened by data.

In addition, some asserted that they noticed that the use of software made a difference for some of their students. In that respect, B3 stated: 'basically you want them to learn something and you think some of them have otherwise will not learn with the chalk and talk'. In the same context, A1 stated: 'I think the use of software could be very helpful. It could solve some of traditional problems that mathematicians face in trying to explain the materials to the students'.

Before moving on to the next point, it should be noted that one of the participants expressed disappointment and surprise at what he had observed by his fellow lecturers of mathematics regarding their reluctance to use software in their teaching, even though computers were created by mathematicians. In this respect, A4 said: 'Why? Something wrong here; software was created by mathematics the queen of sciences, so it is time for mathematics the queen of science to benefit from that, and so people can use this technology in seeing her beauty'.

Software engages students and makes them interested

Engaging, involving and making students interested, or trying to encourage them to have fun when learning mathematics, were all seen as part of the motivational value of the software mentioned by some of in this study. A1 suggested that some students may find 'traditional' ('chalk and talk') lectures boring; software could appeal to these students. A4 described how he tried to give his students the attitude of 'having fun' with mathematics in the 'cyber-classes' he designed for his students, which he used in Moodle for all of his courses. He asserted that his students seemed to enjoy the attitude of learning mathematics with 'fun'. They were interacting with each other and he was sitting with them virtually as one of them watching and observing like 'a guide on the side' trying to minimize his interference. B2 was another lecturer at the other university who described the happiness he saw on his students' faces whenever he used Maple to rotate some of the surfaces and to find an estimate of the volume of some geometric shapes, stressing that students' interaction has actually risen since he started using this package. A2 emphasised that mathematics lecturers in particular should use technology to their advantage to help their students focus their attention on the most important points of lectures. He stated: 'when teaching a highly intellectual discipline such as mathematics; you can put so many things which can make the students happy'. For example, he stated that lecturers could attract students' attention by preparing slides with different fonts and colours which helped students to draw a distinction between theorems, corollaries, lemmas, definitions, examples, remarks and proofs. He felt that this could positively affect lecturers' presence and performance and help in making very well presented lectures.

Software saves lecturers' time in their teaching

Some participants stated that they used software in order to make teaching more efficient with respect to saving time for performing other tasks rather than spending time computing routine calculations. For example, B3 asserted that it was a waste of time if every time he expanded a function in a Fourier series he had to do it manually in front of his students. Thus, in such a situation, he found it useful to let the software compute the expansion in order to save him time for other, probably more urgent, tasks. A2 mentioned an example of his use of PowerPoint presentations as a tool to organize his thoughts and to save time in lectures, especially when he taught early-stage courses at the undergraduate level where students were often slower in their writing. One of the benefits of using software when teaching mathematics that was mentioned by A2 was that the use of software saves lecturers from misleading students because lecturers might make errors unintentionally when they write something randomly on the board.

Vocational and professional preparation

Some lecturers reported that they felt they were obligated, as part of being faithful to their career, to expose their students to the available technology as they might need to use it later in life. For example, B2 stressed the point that every graduate of mathematics must receive minimal exposure to at least one of the major mathematical software packages during their university studies. A2 shared the same view and stated that 'going computational has been the trend in the world nowadays, and all the workforce needs are demanding graduates who are knowledgeable, particularly in dealing with technology and mathematics graduates should be in the forefront when it comes to dealing with technology'.

Easy access to ICT

At both universities covered by this study, there was easy access to ICT facilities without major problems. At both campuses, Internet connections were available everywhere; all classes and labs were equipped with IWBs, multimedia projectors and Internet connections. Both departments had fully equipped self-learning labs. As far as mathematical software packages are concerned, one mathematics department had already subscribed to major mathematical software packages such as MATLAB, Mathematica, Maple and Mathcad; whereas the other department was in the process of subscribing to these packages. Regarding statistical packages, both statistics departments had full access to major statistical packages such as SAS, Minitab, SPSS and STATISTICA. Both campuses had licenses for LMS, namely Bb and Web CT.

It is worth noting here that A4, who was one of the enthusiasts who believed in the importance of using software in teaching mathematics, had overcome the difficulties he faced with respect to his university not having licenses for major mathematical software packages such as Mathematica and Maple, by using free Web-based software that did not require a license, such as GeoGebra and Wolfram Alpha. A4 stated: 'now we are in different stage completely we are. Everything is ready, everything is beautiful. In the old times, I couldn't find labs for my students. Nowadays, each student has his own laptop and wireless connection in any lecture hall. They can set in the coffee room and do their work, so it is a different environment; the environment is beautiful nowadays'.

The university encouraged the use of ICT

Lecturers thought that universities at managerial level were pushing for e-learning. Each had a deanship of e-learning, which gave trainings and provided technical support for lecturers on the effective use of ICT to achieve learning objectives. The academic freedom that university lecturers enjoyed gave them the choice to teach either with or without the use of software. In this regard, A4 stated: 'as far as policy, here we have the chancellor, we have the deanship, the dean in our college, and everyone is trying their best to implement it. But you cannot force people'.

In some courses, it is very applicable or even required to use software

As indicated previously, the use of software was not required nor even strongly recommended in all mathematical courses. According to the participants (e.g. A1, B6), the need to use software in mathematical courses varied depending on the nature of the course. For example, in some mathematical courses, especially in pure mathematics courses based on theory and studying proofs, it was very difficult to use software, except for communication purposes. Other than that, it was left to the lecturer to decide whether or not to use software. However, there were some courses in which the use of software was probably more urgent than in others. For example, the need to use software in a course such as calculus in three-dimensional spaces was perhaps more pressing than in calculus in only one dimension. This is because in such a course, students need to integrate three times, which will produce solid-shaped functions. Therefore, lecturers may need to use software to let their students visualize such complicated functions.

As B6 described, the general practice was that some lecturers used software packages as a supplement to help students understand or illustrate some of the topics. However he stressed that in computational mathematics courses, students were required to use software to compute algorithms, or to write their own codes. B5 pointed out that he could not do numerical analysis without using software, because as he put it, 'it takes forever if you do not use software'. In such courses, it was specified in the syllabus that lecturers must use software, especially software with programming capabilities, such as Mathematica, MATLAB or any general programming languages such as C++ or Fortran. In addition, all statistical courses were by nature required to use statistical software to perform data analysis; as B9 stated: 'you cannot do statistics course without software. We specified in the syllabus you need to use that package'. B1 believed that the use of software was more appropriate when teaching a course for non-specialist mathematics (e.g. engineering students) because as he put it, 'engineering students just need to know the skills; they do not need to learn 'epsilon-delta' and all these details. For mathematics students though it is different philosophy, here we want to strengthen our students' background in the thinking of mathematics not just on doing mathematics'.

It is quite easy to use mathematical software packages

The interviewed lecturers who used software packages stressed unanimously that mathematical packages, particularly symbolic manipulators such as Maple and Mathematica, were easy to learn and use compared to more 'traditional' programming languages such as Fortran or C++. They stressed that the more traditional programming languages were not seen by their students as user friendly since, in such languages, syntaxes and writing programs were sometimes very complicated. In mathematical software packages, however, to plot functions or to perform mathematical calculations, one mostly uses ready-made, built-in functions or very easy syntaxes.

After discussing the rationale for using software in teaching, we will move on to discuss the other side, which deals with the rationale for not using mathematical software packages in the teaching of mathematics at the university level.

Rationale for not using software

Doubts about the value of software will be discussed first.

Presentations should be rough and ready (use of chalk)

Some interviewees explained that when it came to teaching mathematics, it was more natural to use 'chalk and board' as this supported a more fluid and free flowing style of teaching and made it easier for the students to interact and reflect. A2 made it clear that he firmly believed that when it came to teaching mathematics, it was inevitable and unavoidable to use the board, whether a blackboard or a whiteboard, a traditional board or electronic board. He asserted that it was possible in any other subject other than mathematics to replace the board with any other means, such as PowerPoint presentations, but when dealing with mathematical topics, the commentary and explanations on the board were not optional, but rather necessary. He explained when writing notes on the board to start from some logical base and then keep deducing new ideas from old ones, and then exemplifying things. B5 stressed the importance of the time spent writing down the small details of mathematical problems instead of posting them using a PowerPoint presentation. He emphasized that from his experience, when he wanted to save time by posting the problem, thinking that the students had understood what the problem was asking them to do, when they started working on the solution he realised that they had not yet absorbed what the problem was saying and what they were supposed to do to start solving it. However, they had absorbed what the requirements of the problem were when the lecturer wrote down the problem on the board and discussed with the students what was required. He stressed that this amount of time was essential to help the students absorb what the problem was about. He concluded that when lecturers try to save time by using a PowerPoint presentation, they will lose the students.

It is all a 'black box'—overreliance on it

Some of the interviewees expressed their concerns that if students were allowed to use software packages at the university level, which involves learning basic concepts in order to establish students' mathematical backgrounds, this might be counterproductive and harmful, especially at this foundational stage of their journey to learning mathematics. The idea of using software packages as a black box, providing answers without or with little understanding of the mathematical ideas behind the use of the software, was a concern mentioned by more than one of the mathematicians interviewed for this study. B5 questioned the value of using these more advanced software packages such as Mathematica and Matlab when learning mathematics. He stated that in these packages, one statement could do everything and the learner might not know the inner workings of the software and he or she could not easily control it. B1, who was a mathematics lecturer with a managerial responsibility in the department of mathematics B, emphasized this particular point of view, which was deemed to have great importance in his view, by saying: 'later they can just go and forget all about calculus and since they have the packages they can do whatever they like by it, but for now you want to strengthen their background on the thinking of mathematics, not just doing mathematics'. B4 did not agree with such a concern, saying: 'I don't think so because if you look at all the top universities everywhere in the world from Cambridge to Harvard, it is fully interactive you have software packages. Even to use software packages you have to understand you cannot just press button; when you use a software package you have to understand what is you are doing, the nature of the problem, generate a mathematical model and then you have to adjust the parameters to know how to simulate what you want. So mathematical understanding is implied in your use of software—you cannot avoid that'. B4 added another point in this regard, stressing that in the case of over-reliance on technology or if a user did not know how to perform the calculations manually, in fact very quickly he or she will likely find it very difficult to do so using technology or he or she will inevitably face difficulty in subsequent courses.

A4 stressed the distinction between the two approaches in this regard. The first approach, which was 'bad' and 'distractive by all means', was the use of technology as a black box by pressing on a button to get the result without any knowledge of how the inner functions were working or what was happening inside the 'box'. The second approach was the glass-box approach, which could be described as learning through building. In such an approach, students were trained to build the icon that they pressed. He stressed that unless we turn to construction, and to seeing how things work, or at least thinking about how they work. He stressed a very important point: lecturers should be careful not to use the black-box approach unless they carefully pass through the first stage of the glass-box approach. He gave an example of a package such as MATLAB. It should not be used in a course for matrices laboratory courses while students are learning the procedural skills of how to do matrices calculations. There, they must not allowed to use it as a black box, but they should use it to build little programs in MATLAB, then in a linear algebra course, for example, which is an application of that, they should be allowed to use it as a black box because students then know the Gaussian elimination process so they should be allowed to use it and use the lesson time on higher-thought activities. A5 shared the same concerns about the black use of software; he stated that he always warned his students not to resort to using ready-made functions in the software before having full knowledge of the theory behind such functions. For that reason, he always asked his students to write a short programme when solving any problem using the software. In this way, the students would develop an understanding of the theory and the rationale for carrying out the procedures. Perhaps this was why an experienced lecturer interviewed for this study chose to teach numerical analysis, a course which relies entirely on the use of software, using Excel instead of MATLAB or Fortran. B1 asserted that for this reason, and in order to strengthen students' abilities in basic mathematics, the use of calculators, including graphic calculators, was not allowed in any of the calculus courses.

Learning requires effort

There was a view that at the undergraduate level, students need to make additional efforts and spend considerable time to learn mathematics from first principles before resorting to 'ready-made' functions. A7 and B5 shared the view that taking the hard way to learning is the right way, and will produce thoughtful and skilled mathematical minds. They saw learning as a daunting process, at the start of which the learner was probably 'weak', and as time passes his or her mathematical abilities and skills become stronger. They believed that over-reliance on technology, especially in the establishing stage of learners' skills, was resorting to the easy way out, which would weaken the basic mathematical skills of learners and would produce 'lazy' graduates. In respect of visualization, B5 explained that when he was a student, he was initially weak in visualization but became a good 'visualizer' after years of sustained effort.

Now that we have discussed doubts about the value of software, we will explain some of the contextual problems that prevented some lecturers from using software when teaching mathematics courses.

In some courses software is not applicable as they are about theory, not techniques

‘The course nature does not require the use of software’ was the top reason given by some of the interviewees as to why they did not use mathematical software while teaching. They stressed that they simply could not or did not know how to use these packages when the goal, of course, is the production of proofs. Some examples of such courses included: advanced calculus, real and complex analysis, number theory and group and Galois theories. B5 pointed out that, unlike any other subject, teaching undergraduate mathematics aims to build students’ mathematical background, and therefore they must rely heavily on the chalk and board, and limit the use of software as much as possible as it may be counterproductive at this stage. He added that although his specialty is computational mathematics, he shared this view with the mainstream mathematicians. He went on to say: ‘I have not seen any book which would implement, say, Mathematica in learning advanced calculus because in such courses your concentration is on producing proofs where Mathematica doesn’t do that’. B3, who used Mathematica heavily in all of his calculus and differential equations classes, asserted that in some area of mathematics, it was very difficult to integrate technology. For example, it was possible and even desirable to use software when teaching a calculus course for engineering students because the engineering students need only to learn the skills and are not interested in the theoretical side. On the other hand, he stressed that in some courses offered to students specializing in mathematics, for example topology or set theory, in which the focus is on the production and study of proofs, software use is useless.

There is not enough time to prepare other approaches or deviate from the syllabus

Some interviewees (e.g. A4, B1) expressed that with the heavy and fixed content mathematical curriculum and given the limited number of lectures that were allocated to cover all that, there was little time if a lecturer wanted to go in the details of Matlab, Mathematica, etc. This was because the lecturer would then have to devote at least two lectures, or perhaps more, just to cover the basics of the software. Also, B3 stressed that all courses in mathematics departments, especially the undergraduate courses, were coordinated, with probably more than one section having common examinations, which made deviating from the fixed syllabus even more difficult. A2 complained that in his department, most of the time they were not told which course they were going to be teaching before the commencement date of a semester. He added that if he did not know what course he would teach, he could not prepare well-thought-out notes utilizing the available software; such preparation would certainly need more time.

Software use is not assessed

Interviewees from both departments (e.g. A5, B6) stated that the use of software was not assessed. Software was not allowed during examinations in most of the courses offered by the mathematics departments in the two universities. A1 stated that in some of the introductory courses (e.g. calculus courses) students were only allowed to use basic (non-programmable) calculators during examinations.

A5, who was teaching a numerical analysis course, stated that students were not interested in learning about software because they were only required to work on one homework assignment in which the use of software was required. He felt that many students and lecturers would not be interested in using software unless they were 'forced' to do so by including software use in in-class examinations in any form. B5 went further, suggesting that the use of software could not be assessed in class even in computational mathematics courses, because in his view, it was illogical and impractical to ask the students to write a programme during an in-class examination. He felt that it would be very difficult for the student to write a good programme, debug it and expect good results to come out in the time allocated for an in-class exam. He added that the student may fail an exam simply because he or she could not debug the programme on time.

The prevailing culture in mathematics departments is not to use software

In both mathematical departments, there was a general encouragement to use ICT, but a more laissez-faire approach to implementation. In relation the use of software in line with the objectives of the curriculum, direction and support were limited. Lecturers from both departments reported that the prevailing method of teaching was the traditional board-based approach. Some lecturers reported that in their department there was more emphasis on research rather than teaching, since promotions of faculty members depended on the research they produced and did not depend on the quality of teaching they provided, or as B2 put it: 'so would a faculty spends some time thinking how do I improve my teaching style. Well, I will go give my lecture_ who cares? as long as I am doing my research. Why I would care about really working hard to develop whatever thing to prepare my lectures and integrate technology'. B3 reported giving a seminar for the faculty members in his department on how to integrate software into mathematics teaching. He asserted that the attendees were very happy about that seminar and they actually called for another short course. However, he added that there was a difference between being enthusiastic about the use of software in teaching and actually applying it in their lectures, and he emphasized that such changes may need some time, even for those who were very excited about them.

It is difficult for an individual lecturer to design course materials that involve the use of software—there needs to be collaboration

There were calls for more cooperation between lecturers to produce modern curricula that included the use of software. A4 complained that collaboration was somehow missing in the mathematics department where he taught. He stated that everyone was setting in his office and they were not working as a community. A1 drew a distinction between two different things when it comes to using mathematical software. He stressed that the first case that he may not be that enthusiastic about was making a mathematics course, or trying to build interactive lessons using these packages because this would require him to have advanced knowledge of how to produce this material, and it might require things that do not interest him or knowledge that is not purely mathematical. On the other hand, the issue that he was interested in knowing how to solve mathematical problems using various mathematical software packages. For this reason, B3 stressed the need for collaborative work that includes mathematicians who have used the technology and those who have not used the technology for the production of such materials. Also, A4 proposed the idea of creating a common 'cyber-class' for every mathematical course in all universities in the country. He asserted that this was the beauty of e-learning and stressed that it should be a collaborative effort where an individual lecturer was not doing it alone, but he or she was doing it with colleagues. In these cyber-classes, all students in all branches of universities in the country who enrol in this course every week will be studying the same materials, doing the same activities, doing the same assignments and sitting the same examinations. He stressed that this would have better quality management than the current situation where each individual lecturer has to create his or her own cyber-class, because teamwork and collaboration are more important and higher quality than individual work.

Survey Questionnaire

(The final questionnaire was formatted slightly differently to this.)

1. What is your gender?

   □Male

   □Female

2. How many years have you been teaching university-level mathematics (statistics)?

   □1-5

   □6-10

   □11-15

   □16-20

   □20-

3. Please indicate your main research interest (s)

   □Pure mathematics

   □ Applied mathematics

   □ Computational mathematics

   □ Statistics

   □ Others-please specify:

4. On average, how often do you use software in your research?

   □Never

   □Rarely

   □Sometimes

   □Often

   □Always

5. I used software when I was a student:

   □yes

   □no

6. Do you normally teach

   □Mathematics students

   □ Statistics students

   □ Introductory courses for students of different subjects

7. Do you normally teach

   □ First year undergraduate courses

   □second year undergraduate courses

   □third and/or fourth year undergraduate course

8. Which courses are you teaching this year?

   □Calculus □Abstract Algebra □Linear Algebra □Multivariable Calculus □Analysis □Topology

   □Geometry □ODE □PDE □Statistics□ Operations research □Numerical Analysis □Others -please specify:

9. To what extent do you agree or disagree with the following statements:

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
Projection systems are available in most lectures rooms					
It is not difficult for me to schedule a class in a computer lab when I want to					
I have access to the software packages I need for teaching					
I have access to technical support if I need it					

10. Have you ever participated in any technology training provided by your department or elsewhere?

□no □yes at the department □yes university IT services □yes outside the university

I would be happy to attend technology training workshops: □yes □no

11. In a typical academic term, in about which percentage of your lectures do you use mathematical or statistical software?

□Never

□25% or less

□26-49%

□50-74%

□75% or more

12. To what extent do you agree or disagree with the following statements about Mathematical and Statistical software?

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
I find the appropriate software difficult to access in my university					
I think using software in teaching distract students from understanding mathematical concepts					
I don't feel confident using software in my lectures					
I think it is better to use software only in courses that require it					
Software use enables students to become better problem solvers					
Students should focus in introductory courses on mathematics rather than learning to use software					
Students don't like using software for themselves					
The courses I teach do not require the use of software					
Students like to see lecturers use software packages in teaching					
I don't know how to integrate software into my teaching					
Most software is sufficiently user-friendly to be used in classes					
It is difficult to assess what students know if they can use software in tests					
Teaching students how to use software isn't my job					
Teaching is more interactive using software					
Students may use software to provide answers with little understanding of mathematical concepts					
I would like to use software more in my teaching					

I think the use of software in teaching undergraduate maths does more harm than good					
Software allows me to represent concepts in different ways					
I can explain concepts more easily using software					
It is important for students to get hands on practice of software					
Software should be used only to supplement teaching					
Software use helps to focus teaching away from time-consuming calculations					
All math lecturers should use software at times in their teaching					
Students learn math better with software					
It should be left up to the lecturer whether to use software or not					
Software packages are useful for doing mathematics, not learning mathematics					

13. If you never use software packages in your teaching, please say why this is

14. If you use software

How and where I use it

	Never	Occasionally	Frequently	Always
I use software in my lecture room				
I use it in a computer lab				
I assign homework which requires use of software				

I use software for

	Never	Occasionally	Frequently	Always
Visualisation (e.g. creating plots in 2D and 3D and animating them)				
Symbolic manipulations(e.g., derivatives, integrals, solution of linear equations, matrix operations, series operations, polynomials, algebraic simplification, optimizations)				
Numerical computations				
Statistical analysis				
A programing language (i.e. allowing users to implement their own algorithms)				

Please specify-

15. Which of the following do you usually use in your teaching?

Subject software: (Tick as many as apply)

□Matlab □Maple □Mathematica □Mathcad □Scientific Work Place □Spss □Minitab □Sas □Tora □Lingo □spreadsheet software (e.g., Excel)

General software:

□Presentation software (e.g., PowerPoint) □Smart Boar

Learning Management Systems:

□VLE software (e.g., Web CT, Blackboard, or Moodle) for: □Posting lecture notes □Emailing students □Discussion forums □Quizzes □Support material such as past exam papers □setting and submitting online homework □others- please specify:

16. To what extent do you agree or disagree with the following statements

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
Most entry level classes are too large for the available computer labs					
The syllabus in the undergraduate mathematics program limit the use of software					
Only a few of my colleagues are enthusiastic about using software in their teaching					
It takes too long to develop software-related teaching materials					
The syllabus should be modified to include more use of software					
There isn't enough time to incorporate software into math curriculum					
There is too little support for lecturers who want to integrate software in their teaching					
My department encourages the use of software in teaching					
It is not worth using software in classes, because it cannot be used in tests					
There is enough training for lecturers who want to teach with software					
The majority of faculty members within the math department are users of software in their teaching					
Students do not pay attention to the use of software because it is no included in the tests					

17. Which better fits you?

	This is more like me	This is more like me	
Mathematics involves mostly facts and procedures that have to be learned			In mathematics you can be creative and discover things on your own
Students who aren't getting the right answers need to practice on more problems			It doesn't matter whether students get the right answer as long as they understand the math concepts inherent in a problem
Students should construct many of their own math problems			It's important for students to complete assignments exactly as the lecturer planned
Mathematical ability is something that remains relatively 'fixed throughout a person's life			All of my students would be good at math if they worked hard at it
lecturers should facilitate learning, rather than teach directly			lecturers should teach directly, rather than just facilitate

1. Do you have any further comments to make on the use of software to teach mathematics and statistics, can you think of cases in which it can be particularly helpful or unhelpful?

Thank you very much for your participation

Raw findings

Means of delivering the questionnaire

	N (recipients)	N (respondents)
printed version	170	109
online version	251	42
Total	421	151

Table 1: Years of teaching experience in higher education

	N	%
fewer than six years	11	7
6–10 years	28	19
11–15 years	46	31
16–20 years	26	17
more than 20 years	40	27

N = number of respondents; % = percentage of respondents

Table 2: Gender of the respondents

	N	%
male	130	86
female	19	13
undisclosed	2	1

Table 3: Respondents' mathematics specialisms

	N	%
pure mathematics	49	32
applied mathematics	36	24
computational mathematics	33	22
statistics	33	22

Table 4: The respondents' students

	N
mathematics students	115
statistics students	44
introductory courses	83

Table 5: Courses taught

	N
Calculus	72
Differential equations	41
Statistics	37
Linear algebra	34
Numerical analysis	30
Partial differential equations (PDE)	29
Abstract algebra	25
Multivariable calculus	22
Analysis	20
Operation research	12
Topology	9
Geometry	5

Table 6: Use of software in research

	N	%
Never	6	4
Rarely	29	19
Sometimes	34	23
Often	21	14
Always	61	40

Table 7: Use of software as a student

	N	%
yes	122	81
no	29	19

Table 8: Software use in teaching (percentage of classes)

	N	%
never	40	27
25% or fewer	25	17
26–49%	32	22
50–74%	26	17
75% or more	26	17

Table 9: Software use in teaching

	N	%
no/low users	65	44
high users	52	34

Table 10: Where software was used

	N
I use software in a lecture room	101
I assign homework which requires the use of software	79
I use software in a computer lab	64

Table 11: Purposes of software use in teaching

	N
symbolic manipulation	92
visualization	82
numerical computations	64
as a programing language	59
statistical analysis	52

Table 12: Most commonly used software packages in teaching

	N
Matlab	49
Mathematica	48
Excel	36
SPSS	35
SAS	29
Maple	27
Scientific workplace	11
Lingo	9
Mathcad	8
Tora	6

Table 13: Use of general software

	N
presentation software (e.g. PowerPoint)	84
Smart Board	59

Table 14: VLE use

I use the VLE for:	N
posting lectures notes	105
supporting materials, such as past exams	98
Quizzes	62
e-mailing students	51
setting and submitting online homework	37
discussion forums	21

Table 15: Access to software, technical support and computer labs

	N	SD %	D %	Ne %	A %	SA %
Projection systems are available in most lectures rooms	150	2	5	4	67	22
I have access to the software I need for teaching	147	3	11	20	56	10
I have access to technical support	149	2	24	28	41	5
It is not difficult to schedule a class in a computer lab	148	2	19	24	41	14
Most entry-level courses are too large for the available computer labs	139	4	17	22	46	11

Table 16: Appropriate software is difficult to access

	N	SD %	D %	Ne %	A %	SA %
I find the appropriate software difficult to access in my university	141	12	55	18	13	2

Table 17: Technology training

	N	Yes %	No %
Have you ever participated in any technology training provided by your department or elsewhere?	149	77	23
I would be happy to attend technology training workshops	149	91	9

Table 18: Training and support

	N	SD %	D %	Ne %	A %	SA %
There is enough training for lecturers who want to teach with software	139	5	42	30	22	1
There is too little support for lecturers who want to integrate software in their teaching	139	4	23	25	42	6

Table 19: Overreliance on software

	N	SD %	D %	Ne %	A %	SA %
Using software distracts students from understanding concepts	149	19	27	16	19	19
Students may use software to provide answers with little understanding	150	16	16	16	37	15
The focus in introductory courses should be on math rather than on learning to use software	144	16	30	17	21	16
The use of software for teaching undergraduate math does more harm than good	149	17	38	16	15	14
Software packages are useful for doing math, not for learning math	140	8	41	5	35	11

Table 20: Epistemic value of using software

	N	SD %	D %	Ne %	A %	SA %
Software use helps to refocus teaching away from time-consuming calculations	140	1	5	6	69	19
Students learn math better with software	140	2	26	14	44	14
Software use enables students to become better problem solvers	141	2	28	16	43	11

Table 21: Value of software for teaching

	N	SD %	D %	Ne %	A %	SA %
Software allows me to represent concepts in different ways	140	0	6	14	59	21
I can explain concepts more easily using software	139	0	16	17	53	14
Teaching is more interactive using software	140	0	12	12	61	15

Table 22: Motivational value

	N	SD %	D %	Ne %	A %	SA %
Students like to see lecturers using software packages in their teaching	139	1	10	34	48	7
Students don't like using software themselves	141	4	44	31	21	0

Table 23: Curriculum requirements

	N	SD %	D %	Ne %	A %	SA %
The courses that I teach do not require the use of software	140	19	29	6	41	5
The syllabus limits the use of software	139	5	30	11	50	4

Table 24: Modifying the syllabus

	N	SD %	D %	Ne %	A %	SA %
The syllabus should be modified to increase the use of software	138	6	41	12	33	8

Table 25: Assessment issues

	N	SD %	D %	Ne %	A %	SA %
It is difficult to assess what students know if they can use software to do tests	141	5	38	12	42	3
Students do not pay attention to the use of software because it is not included in tests	137	4	30	17	44	5

Table 26: Time constraints

	N	SD %	D %	Ne %	A %	SA %
There isn't enough time to incorporate software into the curriculum	138	5	40	16	36	3
It takes too long to develop software-related teaching materials	139	7	40	15	36	2

Table 27: Confidence and competence in using software

	N	SD %	D %	Ne %	A %	SA %
I don't feel confident using software in my lectures	141	17	31	12	36	4
I don't know how to integrate software into my teaching	141	18	60	10	11	1

Table 28: Intention to use software in teaching

	N	SD %	D %	Ne %	A %	SA %
I would like to use software more in my teaching	141	2	30	15	42	11

Table 29: Departmental influences

	N	SD %	D %	Ne %	A %	SA %
My department encourages the use of software in teaching	139	3	19	34	40	4
Most of the lecturers within the math department use software in their teaching	137	7	62	16	13	2
Only a few of my colleagues are enthusiastic about using software in their teaching	139	3	11	17	60	9

Table 30: How should it be?

	N	SD %	D %	Ne %	A %	SA %
All math lecturers should use software at times in their teaching	140	5	42	18	28	7
It should be left up to the lecturer whether or not to use software	140	3	21	8	58	10
Software should be used to supplement teaching	141	2	16	12	60	10

31: Respondents' research interest broken down by computational focus

	N	%
Non-computational subjects	85	56
Computational subjects	66	44

32: Never used software in the lecture room, broken down by mathematics specialism

	N	%
Non-computational specialists	33	43
Computational specialists	4	7

33: Always used software in the lecture room, broken down by mathematics specialism

	N	%
Non-computational specialists	5	7
Computational specialists	19	31

34: Used software in the computer lab, broken down by mathematics specialism

	N	%
Non-computational specialists	18	23
Computational specialists	46	75

35: Assigned homework requiring the use of software, broken down by mathematics specialism

	N	%
Non-computational specialists	23	30
Computational specialists	56	92

36: Agree or strongly agree with statements about distraction

	Non-computational specialists (%)	Computational specialists (%)
The use of software distracts students	52	21
Software does more harm than good	35	23

37: Disagree or strongly disagree with statements about the value of software

	Non-computational specialists (%)	Computational specialists (%)
Software use enables students to become better problem solvers	36	20
I can explain concepts more easily using software	46	15
Students learn math better with software	42	11

38: Disagree or strongly disagree with the following statements

	Non-computational specialists (%)	Computational specialists (%)
Teaching is more interactive when using software	20	2
Students like to see lecturers use software packages in their teaching	15	5

39: Agree or strongly agree that the syllabus limits the use of software

	Non-computational specialists (%)	Computational specialists (%)
The undergraduate mathematics syllabus limits the use of software	71	33

Table 40: Agree or strongly agree that assessment will be difficult if software is allowed for doing tests

	Non-computational specialists (%)	Computational specialists (%)
It is difficult to assess what students know if they use software to do tests	68	15

Table 41: Agree or strongly agree that there is not enough time to incorporate software

	Non-computational specialists (%)	Computational specialists (%)
There is not enough time to incorporate software into the mathematics curriculum	55	17

Table 42: Agree or strongly agree with the following statement

	Non-computational specialists (%)	Computational specialists (%)
I don't know how to integrate software into my teaching	20	0

Table 43: Disagree or strongly disagree with the following statement

	Non-computational specialists (%)	Computational specialists (%)
I would not like to use software more often in my teaching	41	20

Table 44: Agree or strongly agree the following statement

	Non-computational specialists (%)	Computational specialists (%)
My department encourages the use of software in teaching	28	66

Table 45: Disagree or strongly disagree with the following statement

	Non-computational specialists (%)	Computational specialists (%)
It should be left to the lecturer whether or not to use software	13	38